Learning-based Intrinsic Reflectional Symmetry Detection

Yi-Ling Qiao, Lin Gao*, Shu-Zhi Liu, Ligang Liu, Yu-Kun Lai, and Xilin Chen*

Abstract—Reflectional symmetry is a ubiquitous pattern in nature. Previous works usually solve this problem by voting or sampling, suffering from high computational cost and randomness. In this paper, we propose a learning-based approach to intrinsic reflectional symmetry detection. Instead of directly finding symmetric point pairs, we parametrize this self-isometry using a functional map matrix, which can be easily computed given the signs of Laplacian eigenfunctions under the symmetric mapping. Therefore, we manually label the eigenfunction signs for a variety of shapes and train a novel neural network to predict the sign of each eigenfunction under symmetry. Our network aims at learning the global property of functions and consequently converts the problem defined on the manifold to the functional domain. By disentangling the prediction of the matrix into separated bases, our method generalizes well to new shapes and is invariant under perturbation of eigenfunctions. Through extensive experiments, we demonstrate the robustness of our method in challenging cases, including different topology and incomplete shapes with holes. By avoiding random sampling, our learning-based algorithm is over 20 times faster than state-of-the-art methods, and meanwhile, is more robust, achieving higher correspondence accuracy in commonly used metrics.

Index Terms—Mesh Processing, Symmetry Detection, Deep Learning, Intrinsic Reflectional Symmetry, Laplacian, Eigenanalysis

1 Introduction

Symmetry is a common pattern that appears ubiquitously in the world. The majority of living things, including humans (Figure 1), animals, and plants (e.g. flowers) have some form of symmetry. It is also a widely employed design principle in man-made objects, including buildings, furniture, vehicles, to name a few.

Due to its wide applicability, symmetry patterns have been exploited in many tasks, including shape matching [1], segmentation [2], editing [3], completion [4], and understanding [5]. In these application systems, symmetry detection is usually an integral component, so efficient symmetry detection has significant benefits, e.g., to avoid impeding real-time performance in 3D acquisition/reconstruction, and for improved user experience in interactive shape editing by reducing users' waiting time.

To study symmetry, researchers mainly focus on the spatial domain, including extrinsic symmetry defined in Euclidean space or intrinsic symmetry defined in non-Euclidean (manifold) space. Extrinsic symmetry refers to shape invariance w.r.t. rigid (including reflectional) transformations. Compared to extrinsic symmetry, intrinsic symmetry is more difficult to detect due to its much larger solution space, as discussed in the previous work [6], [7], [8].

Given a shape model, intrinsic symmetry detection aims to estimate a self-homeomorphism on the manifold that preserves the geodesic distance between each point pair. Usually, the manifold is discretized as a triangle mesh, and algorithms predict a point-wise correspondence matrix to represent symmetric pairs. State-of-the-art methods for intrinsic symmetry detection is largely based on embedding the symmetry space to some lower-dimensional spaces, such as Möbius transformation space [7], Global Point Signature (GPS) space [8], or functional map space [6], [9] and performing random sampling or voting, which suffers from high computational cost and uncertainty of results due to randomness.

Despite great effort, efficient and robust detection of intrinsic symmetry remains challenging. Existing state-of-the-art methods typically take several seconds or longer to analyze one shape [9], and may produce unreliable results for difficult cases. To address this, we design the first learning-based intrinsic symmetry detection method to handle the intrinsic symmetry problem. Like most existing works, we focus on intrinsic reflectional symmetry as it is most common in the real world. Learning intrinsic symmetry directly on meshes is challenging, due to their irregular connectivity, and the global nature of symmetry. We simplify this problem when designing the neural network, such that it does not directly process the edges and faces of the mesh, but instead takes intrinsic features as input.

Similar to [9], given an input mesh, the symmetry mapping defined on it can be represented using a functional map, or equivalently using a functional map matrix. Laplace-Beltrami eigenfunctions can be extracted to provide a basis for analysis. In the matrix, entries corresponding to eigenfunctions associated with non-repeating eigenvalues are determined by the sign (odd or even) of the eigenfunction after the symmetry mapping is applied. State-of-the-art work [9] determines the sign of the eigenfunction through random sampling. Although it is faster than previous methods, this sampling-based method is still slow (requiring several seconds for a typical mesh), and may not be sufficiently

* *Corresponding Authors are Lin Gao and Xilin Chen.*

- *Yi-Ling Qiao is with the University of Maryland, College Park, USA, and also with the University of Chinese Academy of Sciences, Beijing, China. E-mail: yilingq@umd.edu*
- *Lin Gao and Shu-Zhi Liu are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China. E-mail: {gaolin, liushuzhi16}@ict.ac.cn*
- *Yu-Kun Lai is with School of Computer Science and Informatics, Cardiff University, Wales, UK. E-mail: LaiY4@cardiff.ac.uk*
- *Ligang Liu is with School of Mathematical Sciences, University of Science and Technology of China, Hefei, China. E-mail: lgliu@ustc.edu.cn*
- *Xilin Chen is with Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China. Email: xlchen@ict.ac.cn*

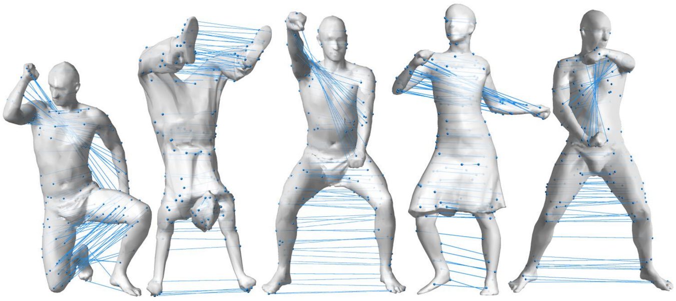


Fig. 1. Intrinsic symmetry on human bodies, detected by our method.

robust.

In comparison, sign patterns can be easily recognized by a human at a glance (see Figure 3). Inspired by this, we label a training dataset for eigenfunction signs and train a neural network to learn such patterns. We design SignNet, a neural network for sign prediction, that in addition to the eigenfunction to be predicted, also takes the first few Laplacian eigenfunctions as input, which effectively encode intrinsic descriptions of the mesh characteristics, while avoiding coping with mesh connectivity explicitly. After predicting eigenfunction signs as entries of the functional map matrix, we apply post-processing to further fine-tune the results (addressing issues such as near-identical eigenvalues and slight non-isometry) and convert the functional map to one-to-one point correspondence. The data and code is available in the repository https://github.com/YilingQiao/intrinsicSym.git.

The main contributions of this work are:

- We propose the first learning-based method to detect global intrinsic reflectional symmetry of shapes. Compared to previous works, our method is much more efficient than state-of-the-art (over 20 times faster). Our method also achieves higher accuracy and is more robust.
- To compute the entries of the functional map matrix, we labeled a dataset containing 3,000 eigenfunctions along with corresponding signs, and design a novel neural network to predict the sign of each eigenfunction. The data and code for annotation, training, and visualization will also be publicly released.

2 Related Work

We now review papers most related to our work, namely those on intrinsic symmetry detection and 3D shape analysis with deep learning.

2.1 Intrinsic Symmetry Detection

Many previous works cast their attention on intrinsic symmetry detection tasks. Ovsjanikov et al. [8] formulate the concept of intrinsic symmetry. They propose to use the Global Point Signature (GPS) [10] to transform the intrinsic symmetry of shapes into the Euclidean symmetry in the signature embedding space. The symmetry is detected by first deciding the sign sequence of eigenfunctions and then finding the nearest neighbors of the GPS of points. Xu et al. [11], [12] extend the concept of intrinsic symmetry and introduce partial symmetry where some parts of an object are symmetric. In this paper, we focus on global intrinsic symmetry due to its wide applicability, as most research in this area does.

To address the large solution space, some works parametrize intrinsic symmetry to some lower dimensional space. A highly related problem is investigated by Mitra et al. [13] who propose a method to symmetrize imperfectly symmetric objects. They find intrinsically symmetric point pairs by voting, and then parametrize possible transformations in a canonical space and optimize the transformation to align symmetric pairs. Kim et al. [7] use another parametrization of symmetry transformations. They find a set of symmetric points by detecting critical points of the Average Geodesic Distance (AGD) function, and generate candidate anti-Möbius transformations that can describe the symmetric transformation by enumerating subsets of the points. As a voting-based method, the running-time could be an issue. Also, the use of anti-Möbius transformation limits the method to handle genus-zero manifolds. Lipman et al. [14] detect symmetry by finding the orbit of points under symmetric transformations. A fuzzy point-wise symmetry correspondence matrix is generated randomly, based on which they further compute a Symmetry Factored Embedding (SFE) and Symmetry Factored Distance (SFD). However, the computation of the correspondence matrix is very time-consuming. Raviv et al. [15] formulate full and partial symmetries and solve the symmetries through a numerical framework. [16] proposes a matching algorithm which can also detect intrinsic symmetries in order to eliminate the symmetric ambiguity in the correspondence between two shapes. They achieve this by minimizing a linear objective function for the sign of eigenfunctions. ZoomOut [17] proposes a general method to efficiently refine mapping and correspondence, which can also be used to optimize intrinsic symmetries.

The relationship between symmetry groups and matrices is studied in [14]. Similarly, Wang et al. [6] establish a homeomorphism between the symmetry group and the multiplication group of matrices. They introduce the functional map to parametrize the symmetry and limit the search space of matrix entries to the subspace of eigenfunctions. However, due to the noise in manifolds and errors during the numerical calculation, eigenvalues which are ideally identical are usually calculated as different values in practice, making it difficult to determine true subspaces and resulting in poor symmetry detection. As described in [6] the continuity and sparsity make functional maps a suitable representation for correspondence problems, including intrinsic symmetry. Functional maps are also used in the work [9]. As also mentioned in [8], eigenfunctions are invariant under self-isometry, apart from sign ambiguity, and the diagonal entries of the functional map matrix are related to the sign of corresponding eigenfunctions. To decide the signs, landmark symmetric point pairs and the geodesic lines connecting them are selected. Nagar and Raman [9] design an explicit solution to this problem, but since their method depends on the landmark pairs, the random sampling requires a trade-off between robustness and computation complexity.

Compared to state-of-the-art methods, our learning-based method avoids explicit sampling and is much faster (over 20 times faster for a typical example). It circumvents the randomness of sampling, and is thus more robust and accurate.

2.2 Shape Analysis with Deep Learning

Our method learns the properties of eigenfunctions defined on manifolds using neural networks. We review research that successfully apply neural networks in tasks related to 3D shapes [18],

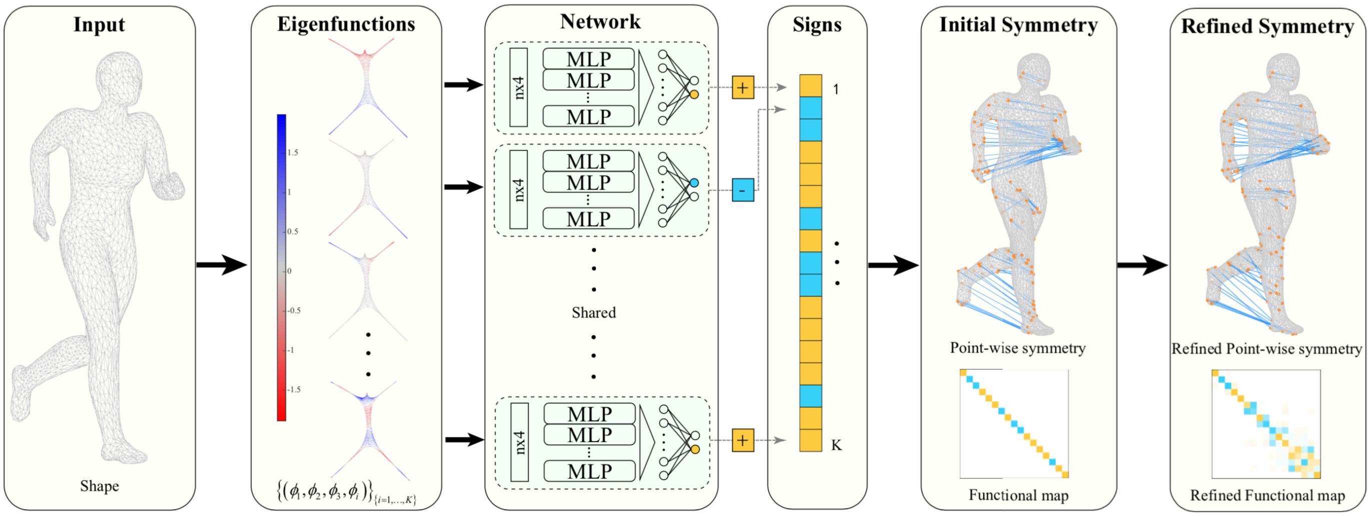


Fig. 2. Pipeline of our method. Given a triangle mesh of a shape, our method can predict point-wise intrinsically symmetric correspondences. First, we compute the Laplace-Beltrami eigenvectors of the mesh. Then we retain the eigenvectors associated with the first K smallest eigenvalues (excluding eigenvalue 0) for analysis. We train a neural network called SignNet, that predicts the sign of each eigenfunction ϕ_i $(i = 1, 2, \dots, K)$, under reflectional symmetry transformation T, i.e., whether $\phi_i \circ T = \phi_i$ or $\phi_i \circ T = -\phi_i$. Instead of feeding the extrinsic positions as input, we use as input the first t eigenvectors along with the i-th eigenfunction ϕ_i, which are invariant under isometric transformation. The second box visualizes the shapes using the first three eigenvectors (ϕ_1, ϕ_2, ϕ_3) as the coordinates and ϕ_i as the color (where blue is small and red is large). The output is a two-dimensional vector, indicating the sign. We later combine K signs and convert them to a diagonal functional map matrix whose entries c_{ii} are either $+1$ or -1. This induces an initial point-wise symmetry map. To correct errors led by repeating eigenvalues and imperfect symmetry, we refine the diagonal matrix and obtain a better symmetry mapping as shown in the last box through post-processing.

[19], [20], [21], [22]. Boscaini et al. [23] design an anisotropic convolutional neural network to learn correspondences across shapes. Masci et al. [24] also design a network in the spatial domain.

Alternatively, another category of work constructs neural networks in the spectral domain. Bruna et al. [25] introduce a spectral convolutional layer on graphs, which can be viewed as a general form of meshes. As described in [26], a fundamental problem of spectral convolution is its dependency on the basis, making it difficult to be generalized to different domains. To mitigate this, Yi et al. [27] propose a network architecture to synchronize the spectral domains and then perform convolutional operations on it. Rodolà et al. [28] design the first network for finding correspondence between shapes using functional maps; however, it does not address the symmetric ambiguity explicitly. Roufosse et al. [29] design fully connected networks to learn features that can generate functional map matrices; however, fully connected networks may suffer from overfitting, and they do not learn on the functional space. Donati et al. [30] propose to learn functional maps in an unsupervised manner, but it can not solve the symmetries of shapes. Li et al. design OptCuts [31] to refine the non-rigid functional map results, which can be adopted as a post-processing step in tasks like correspondence and symmetry detection. Halimi et al. [32] propose an unsupervised method to learn the correspondence between non-rigid shapes, but it is not designed for symmetries.

3 Representing Intrinsic Symmetry by Functional Maps

To cope with discrete and high-dimensional point-wise correspondence matrices, we use functional maps to represent the self-mapping. The functional map was introduced in [33], first used to describe the correspondences between two shapes. In our problem, a self-isometry T can also be viewed as a self-mapping $T : \mathcal{M} \to \mathcal{M}$ on manifold $\mathcal{M}$, which naturally introduces a bijective transformation $T_f \in GL(L^2(\mathcal{M}))$ in the square-integrable space $L^2(\mathcal{M})$, such that

$$\forall f \in L^2(\mathcal{M}), m \in \mathcal{M}, T_f(f)(m) = f(T(m)). \tag{1}$$

Assume that $L^2(\mathcal{M})$ is equipped with an orthogonal basis $\{\boldsymbol{\phi}_i\}_{i=1,2,\dots}$. For each T, the functional map can be represented by a matrix C, with entries $c_{ij} =< T_f(\boldsymbol{\phi}_i), \boldsymbol{\phi}_j >$. For each function $f = \sum_i b_i \boldsymbol{\phi}_i \in L^2(\mathcal{M})$ with coefficient vector $\boldsymbol{b} = (b_1, b_2, \dots)$, the coefficient vector of map $T_f(f)$ is $C\boldsymbol{b}$. In this way, we can represent the mapping by the matrix $\boldsymbol{C}$.

Following the choice of [33], we use the eigenfunctions of the Laplace-Beltrami operator as the basis. For a mesh with N vertices, the discrete Laplacian operator on the mesh is defined as an $N \times N$ matrix [34]

$$\boldsymbol{L} = \boldsymbol{A}^{-1}(\boldsymbol{D} - \boldsymbol{W}), \tag{2}$$

where $\boldsymbol{A} = diag(a_1, \dots, a_N)$ contains vertex weights, with a_i equal to the Voronoi area of the vertex (i.e., a third of the sum of one-ring neighborhood areas). $\boldsymbol{W} = \{w_{ij}\}_{i,j=1,\dots,N}$ is the sparse cotangent weight matrix, $\boldsymbol{D}$ is the degree matrix which is a diagonal matrix with diagonal entries $d_{ii} = \sum_{j=1}^{N} w_{ij}$.

The aforementioned eigenfunction basis $\boldsymbol{\phi} = \{\boldsymbol{\phi}_i\}_{i=1,2,\dots,N}$ is the solution of $\boldsymbol{L\phi} = \boldsymbol{\Lambda\phi}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix whose diagonal entries are eigenvalues in ascending order, $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_N$. For efficiency and robustness, we take the eigenfunctions corresponding to the first K smallest eigenvalues ($K << N$). Note $\lambda_0 = 0$ and the corresponding trivial eigenfunction is ignored.

4 Method

4.1 Overview

Our goal is to detect the intrinsic symmetry of shapes. An intrinsic symmetry is the self-homeomorphism of a smooth surface $\mathcal{M}$,

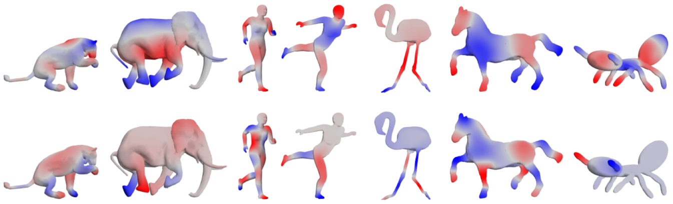


Fig. 3. The Laplace-Beltrami eigenfunction maps on shapes. In this figure, we visualize the eigenfunctions on shapes from SHREC 2007 [35], elephant, and flamingo [36]. Red vertices represent positive values of eigenfunctions and blue values are negative values. First row shows eigenfunctions where $\phi_i \circ T = \phi_i$ (i.e. positive cases), and the second row presents negative cases, satisfying $\phi_i \circ T = -\phi_i$. Obvious symmetric/antisymmetric patterns can be observed in these non-repeating eigenfunctions. We further develop a neural network to predict the signs.

written as $T : \mathcal{M} \rightarrow \mathcal{M}$, which preserves geodesic distances d_g

$$\forall m, n, T(m), T(n) \in \mathcal{M}, d_g(m, n) = d_g(T(m), T(n)). \quad (3)$$

Instead of directly computing a point-wise correspondence matrix, we use a functional map to describe this self-mapping. The functional map defined on the Laplacian basis is represented as a matrix, which is the coordinate transformation matrix w.r.t. the source and target bases. Since the Laplace-Beltrami operator is invariant under isometric transformation, the eigenfunction space stays invariant under self-mapping. Therefore, the matrix $\boldsymbol{C}$ corresponding to the self-mapping T is a block diagonal matrix. More specifically, only one of the two cases holds for eigenfunction $\boldsymbol{\phi}_i$ associated with non-repeating eigenvalues (see also in [8]):

- $\boldsymbol{\phi}_i \circ T = \boldsymbol{\phi}_i$, where $\boldsymbol{\phi}_i$ is called *positive*.
- $\boldsymbol{\phi}_i \circ T = -\boldsymbol{\phi}_i$, where $\boldsymbol{\phi}_i$ is called *negative*.

Therefore, the entry c_{ii} in the matrix corresponding to each non-repeating eigenfunction $\boldsymbol{\phi}_i$ should be either +1 or -1, depending on whether $\boldsymbol{\phi}_i$ is positive or negative. Figure 2 shows the pipeline of our method. We train a network called SignNet to distinguish the sign of eigenfunctions under reflectional symmetry. To provide sufficient guidance, we train the network in a supervised fashion with our annotated data. Given an input shape, once the signs of Laplacian eigenfunctions are predicted using our SignNet, we can fill in the diagonal of the initial functional map matrix $\tilde{\boldsymbol{C}}$ with +1 and -1. However, most of the time the intrinsic symmetry is imperfect, where some areas experience non-isometric deformation. Moreover, there could also be eigenfunction spaces associated with repeating eigenvalues, in which condition the diagonal matrix cannot fully express the mapping. Therefore we use a postprocessing step to fine-tune the initial matrix $\tilde{\boldsymbol{C}}$ to obtain the final matrix $\boldsymbol{C}$.

4.2 Learning Intrinsic Symmetry

We now discuss the steps involved in learning intrinsic symmetry using our method.

4.2.1 Diagonal entries of the functional map matrix

As described in Section 3, we detect intrinsic symmetry by computing the functional map matrix $\boldsymbol{C}$. Although the dimension of $K \times K$ functional map matrix $\boldsymbol{C}$ is already much lower than the $N \times N$ point-wise correspondence matrix, predicting the full $K \times K$ matrix is still challenging for optimization methods or neural networks since there are still too many variables. We further utilize the sparse structure of the symmetry functional map to make it much easier to predict the mapping.

For an intrinsic mapping T defined in Equation (3) and a Laplacian eigenfunction $\boldsymbol{\phi}_i$ associated with a non-repeating eigenvalue λ_i, $\boldsymbol{\phi}_i \circ T = \pm\boldsymbol{\phi}_i$. Generally, the entries of transformation matrix are $c_{ij} = < T_f(\boldsymbol{\phi}_i), \boldsymbol{\phi}_j >$. If all eigenvalues are non-repeating, then the entries of $\boldsymbol{C}$ can be computed by $c_{ij} = s_i$, if $i = j$, or 0 otherwise. More detailed proofs can be found in [8].

This means that $\boldsymbol{C}$ is a block diagonal matrix, where the diagonal entry associated with the non-repeating eigenvalue λ_i is s_i.

4.2.2 Predicting the sign of eigenfunctions

So the problem is much simplified and disentangled, such that we can derive the whole matrix by separately considering the sign of each eigenfunction. The visualization of the eigenfunctions on shapes is shown in Figure 3. From the figure it can be seen that symmetric patterns are rather obvious: positive functions appear symmetric under reflectional symmetry, while negative ones are skew-symmetric. Nagar and Raman [9] propose a sampling-based method to decide the sign of the function. However, this approach depends on random samples and computation of geodesic distances, which takes a long time to compute and may occasionally fail. In this paper, we propose to train a neural network to learn the sign of eigenfunctions.

Figure 2 illustrates the pipeline of our method. Given an input shape, we first compute its Laplacian matrix and the first K eigenfunctions (excluding the trivial eigenfunction associated with eigenvalue 0). Instead of taking the whole shape along with the eigenfunctions as input, which requires the neural network to deal with irregular mesh connectivity, our neural network (SignNet) processes each eigenfunction $\boldsymbol{\phi}_i$ *separately*. Assuming the i-th eigenfunction $\boldsymbol{\phi}_i$ is being processed, the input to the network includes not only $\boldsymbol{\phi}_i$, but also the first t eigenfunctions $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_t$, which capture the characteristics of the input mesh and are also intrinsic.

The output of SignNet is a 2-dimensional softmax vector. The distributions of the eigenfunctions on the mesh can reflect the pattern of the sign to a great extent. Here we do not use the original positions of vertices as input since they are extrinsic features. In contrast, the first t dimensions of Laplacian eigenvectors are intrinsic, thus more suitable for detecting intrinsic symmetry.

To visualize this, in the second block of Figure 2, we plot the embedding of vertices taking the first three eigenfunctions $(\boldsymbol{\phi}_1(p), \boldsymbol{\phi}_2(p), \boldsymbol{\phi}_3(p))$ evaluated at vertex p as vertex coordinates and $\boldsymbol{\phi}_i(p)$ as the color (blue to red means small value to large value). It can be observed that the shapes of the embedding are extrinsically symmetric even if the mesh is only intrinsically symmetric. Also, we can see that those eigenfunctions are either symmetric or skew-symmetric, corresponding to positive or negative eigenfunctions.

In the SignNet neural network, we use Multi-Layer Perceptrons (MLPs) to extract vertex features with increasing complexity. Then a max-pooling is applied on all vertices to aggregate global features. Following the pooling layers are several fully-connected layers with decreasing numbers of channels. In the end, the network predicts a two-dimensional score vector $\boldsymbol{v}_i = (v_{i,1}, v_{i,2})$, i.e.

$$\boldsymbol{v}_i = SignNet(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_t; \boldsymbol{\phi}_i), \quad (4)$$

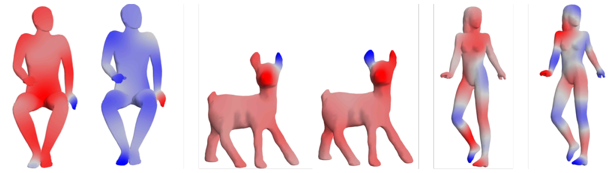


Fig. 4. Eigenfuntions with repeated eigenvalues are neither odd nor even. Those eigenfunctions are labeled as ‘0’ and discarded in the later training process.

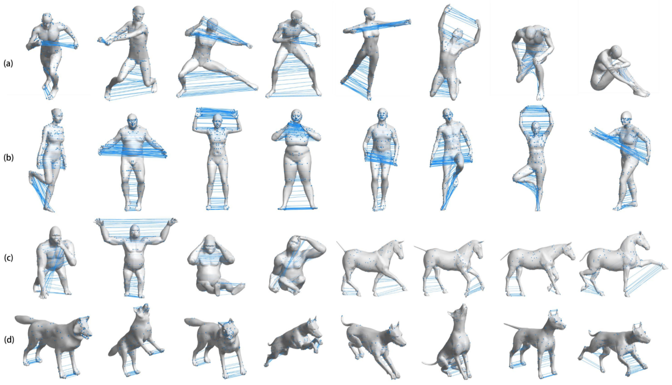


Fig. 5. Our method generalizes well to various shapes including male, female, gorilla, horse, wolf, and dog [38], [39].

such that the sign is predicted to be negative if $\arg\max_k v_{i,k} = 1$, or positive if $\arg\max_k v_{i,k} = 2$, for $k = 1, 2$. Let $\hat{\boldsymbol{s}}_i$ be a two-dimensional vector, $\hat{\boldsymbol{s}}_{\boldsymbol{i}} = (1, 0)$ if $s_i = -1$, and $\hat{\boldsymbol{s}}_{\boldsymbol{i}} = (0, 1)$ if $s_i = 1$. The loss function is designed as the cross entropy between $\boldsymbol{v}_i$ and ground-truth sign label $\hat{\boldsymbol{s}}_i$, formulated as

$$Loss = CrossEntropy(\boldsymbol{v}_i, \hat{\boldsymbol{s}}_i). \tag{5}$$

4.3 Training Data

Our learning-based approach requires a dataset for training. For this purpose, we choose as a training set a fusion of sets SHREC 2007 [35], elephant, and flamingo [36]. This dataset contains deformed shapes which are intrinsically symmetric. As a shape retrieval dataset, SHREC dataset includes shapes of different categories. Meanwhile, they are also independent from the test sets (SCAPE [37] and TOSCA [38]). This ensures fairness and evaluates the generalizability of our learning-based approach. Our method thus generalizes well to various kinds of shapes and poses as in Figure 5.

We built a simple user interface to visualize and manually label each Laplacian eigenfunction as either positive, negative or neither under reflectional symmetry transform. In the labeling process, we first compute the eigenfunctions and visualize them on the meshes (similar to Figure 3 and Figure 4). As a human, we can easily differentiate the “sign” of an eigenfunction in a short time. For example, eigenfunctions in the first row in Figure 3 are symmetric (even) and labeled as ‘+1’, while those in the second row are anti-symmetric (odd) and labeled as ‘-1’. There are also some eigenfunctions that are neither even nor odd as shown in Figure 4. Usually, those eigenfunctions are only *partially* even/odd. We label them as ‘0’ and do not use them for training. The dataset is released to the community to facilitate future research.

4.4 Network Architecture

In our SignNet, the input placeholder is set to work with 4500 points, which are padded with 0 if the mesh has less than 4500 vertices, and for meshes with more than 4500 vertices, they are downsampled to 4500 points. As shown in Fig. 2, we use multi-layer perceptrons (MLPs), max-pooling layers, and fully connected layers. There are five MLP layers, having 64, 128, 256, 512, 4096 channels respectively, and there are ReLU activation layers and batch normalization layers right after the output of each MLP layer. Then we use a max-pooling layer to aggregate the global features. Such a combination of shared-weight MLP layers and max-pooling layers are proven to be effective to fit functions defined on the point set (see the appendix in [40]). Then, four fully connected layers are applied to the global features. Their output channels are 512, 128, 32, 2. The first three layers are also connected with ReLU activation, batch normalization, and (70%) dropout layers.

Although both [29] and our method use deep learning models and the concept of functional maps, the problem settings and the input/output/structure of the network are all different. Their approach cannot be naively used to solve the intrinsic symmetry problem. Considering the design of the network, compared to their fully-connected network, our share-weight MLP network defined on the eigenfunction space uses fewer parameters and is invariant under different input vertex orders.

4.5 Post-processing

Most of the time, the meshes that we are processing are not perfectly intrinsically symmetric. The entries of functional matrices would not be exactly -1 and +1. Moreover, owing to the imperfect triangulation and discretization of the Laplacian operator, in numerical computation, eigenvalues are mostly non-repeating, but there are actually eigenfunction spaces with multiple eigenfunctions. Therefore, the entries associated with sub-eigenfunction spaces need more entries, usually in the form of an orthogonal sub-matrix, to describe the functional map.

In consideration of the above two reasons, we use a post-processing step to correct the matrix and convert it to point-wise correspondence. Similar to [9], we use functional constraints [33] to align the correspondences. As shown in Figure 7, this can correct initially imprecise mappings.

5 Results and Evaluation

We first describe the implementation details of our method in Section 5.1. In Section 5.2 we compare our method with existing methods, both qualitatively and quantitatively. In addition to the accuracy of symmetry, we also measure the run time of different methods, showing the significant superiority of our method in efficiency. We further test the robustness of our method in Section 5.4. Due to the shared-weight structure of our network, our method stays robust under different topology and vertex numbers.

5.1 Implementation Details

We now present details of the training and test process of our SignNet.

The computation of the Laplacian matrix and eigenvectors are described in Section 3. Please refer to [34] for more implementation details related to these steps. We implement the neural

TABLE 1
Comparison of correspondence rate, mesh rate, and running time on the SCAPE dataset. We compare our method with MT [7], BIM [42], OFM [43], GRS [6], ZO [17], and FA [9].

	MT	BIM	OFM	GRS	ZO	FA	Ours
Corr. Rate(%)	82.0	84.8	91.7	94.5	94.2	97.5	97.9
Mesh Rate(%)	71.8	76.1	97.2	98.6	93.05	100	100
Time(s)	18.0	304.26	50.70	20.28	51.58	6.77	0.24

TABLE 2
Correspondence rate (%) comparison on TOSCA.

	MT	BIM	OFM	GRS	ZO	FA	Ours
Cat	66.0	93.7	90.0	96.5	94.6	95.6	96.0
Centaur	92.0	100	96.0	92.0	92.0	100	100
David	82.0	97.4	94.8	92.5	96.5	96.2	97.2
Dog	91.0	100	93.2	97.4	96.8	98.8	100
Horse	92.0	97.1	95.2	99.5	100	97.3	96.4
Michael	87.0	98.9	94.6	91.4	94.7	96.5	98.7
Victoria	83.0	98.3	98.7	95.5	92.8	96.2	97.8
Wolf	100	100	100	100	100	100	100
Gorilla	-	98.9	98.9	100	100	100	100
Average	85.0	98.0	95.1	94.5	96.4	97.8	98.1

network architecture with Tensorflow. The network is optimized using Adam [41] solver. The initial learning rate is set to 1×10^{-4} and momentum is 0.9. We choose to truncate at first 12 lowest eigenvectors (i.e., $K = 12$), and by default, the input feature has 4 dimensions, composed of the first 3 eigenvectors (i.e., $t = 3$) and the i-th eigenvector. We train the network for 150 epochs on a PC with an NVIDIA 1080TI GPU and an Intel i7-7700 CPU. The network inference time is 0.06s. When we train the network using batch size = 4, the GPU memory used is 5.8G. The preprocessing time for computing eigenvectors is 0.03s, and the postprocessing time to finetune the results is 0.24s. In total, our method is still at least 20 times faster than the previous methods.

5.2 Comparison of Results

As one of the biggest advantages of learning-based methods, our algorithm runs much faster than previous sampling based intrinsic symmetry detection algorithms. Also, the neural network can learn some common properties of eigenfunctions across models to distinguish the sign of eigenfunctions. This would avoid randomness of sampling, so also has better performance in terms of correspondence accuracy. In this section, we compare our method with state-of-the-art methods including MT [7], BIM [42], OFM [43], GRS [6], ZO [17], and FA [9] in the following two metrics, widely used in the literature:

1) *Correspondence rate*: Assume that $(m, m'), m, m' \in \mathcal{M}$ is a ground truth correspondence pair, and the algorithm's prediction is $(m, T(m))$. If the geodesic distance $d_g(m', T(m))$ between m' and $T(m)$ is less than the threshold $\sqrt{\frac{area(\mathcal{M})}{20}}$, then we count this point as a correct matching. Correspondence rate measures the ratio of labeled points that are correctly matched.
2) *Mesh rate*: It measures the percentage of meshes whose correspondence rate is above the threshold β. We use $\beta = 75\%$, the same as [6], [9].

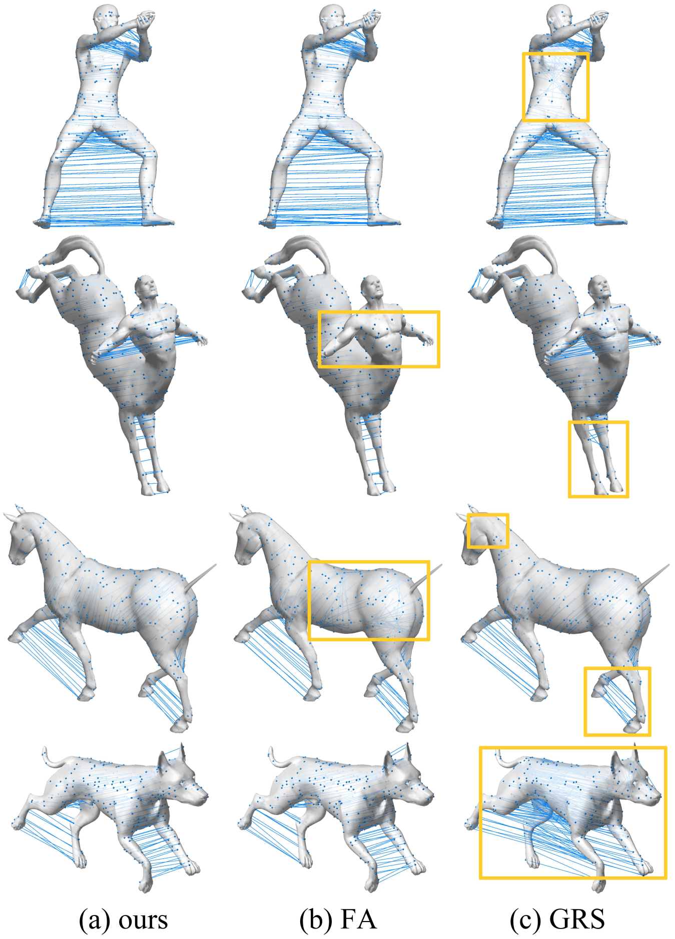


Fig. 6. Qualitative comparison with previous work. (a) is symmetry predicted by our method; (b) is from FA [9]; (c) is result of GRS [6]. We can see that our method has the least artifacts when detecting symmetry.

We run experiments on SCAPE [37] and TOSCA [38] datasets which contain intrinsically symmetric meshes, and the ground truth symmetric correspondences are from [44]. We also test our method on Handstand, Swing [45] and FAUST [39] datasets for qualitative evaluation, as no ground truth correspondences are available. As we mentioned in Section 4.3, our training set is independent of the test sets, to ensure fairness.

The results on the SCAPE dataset of deformed human shapes are reported in Table 1. As can be seen, our method achieves the best accuracy: improving the correspondence rate from the previous best 97.5% (FA) to 97.9%. Both our method and FA achieve 100% mesh correct rate. In terms of runtime, our method is over 20 times faster than FA, and even more than other existing methods.

The results on the TOSCA dataset are reported in Tables 2 and 3 for the comparisons of correspondence rate and mesh rate, respectively. We report performance on individual object categories and the overall average. Our method has similar improvements compared with existing methods. Some qualitative comparison is shown in Figure 6.

5.3 Evaluation of Design Choices

As we said before, by default we use the first three Laplacian eigenfunctions as the coordinates to embed vertices into an intrinsic space. Compared to Laplacian eigenfunctions, the raw positions are not invariant under global rigid transformation, nor under non-rigid isometric deformation, so not suitable for predicting the sign of eigenfunctions on the mesh. In this experiment, we compare using the positions $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\phi}_i)$ versus eigenfunctions

TABLE 3
Mesh rate (%) comparison on TOSCA.

	MT	BIM	OFM	GRS	ZO	FA	Ours
Cat	54.6	90.9	90.9	100	90.9	100	100
Centaur	100	100	100	100	100	100	100
David	57.1	100	100	100	100	100	100
Dog	88.9	100	88.9	100	89.9	100	100
Horse	100	100	87.5	100	100	100	100
Michael	75	100	100	100	100	100	100
Victoria	63.6	100	100	100	91.7	100	100
Wolf	100	100	100	100	100	100	100
Gorilla	-	100	100	100	100	100	100
Average	76	98.7	92.6	100	96.3	100	100

$(\phi_1, \phi_2, \phi_3, \phi_i)$ as input. Table 4 lists the average accuracy of sign prediction on TOSCA and SCAPE datasets. It shows that the accuracy using the position (denoted as *Pos.*) is much lower than that of our design. During experiments, we observe that when models have scales in a large range, the network with position input performs even worse.

We compute the functional map matrix by independently predicting the sign of eigenfunctions. This strategy circumvents the flip of signs and permutation of eigenfunctions. To show the advantage of this strategy, we design another network which takes all the eigenfunctions as input and predicts the whole K diagonal entries at once. We denote this alternative design as *Diag.* in Table 4. We can see the accuracy of sign prediction is much lower than ours. This is probably because the input and output dimensions are too high for the network to learn.

The input to the network is the first t eigenfunctions as well as the i-th eigenfunction, i.e., $[\phi_1, ..., \phi_t; \phi_i]$. Too small the t value would make different vertices indistinguishable, impossible to determine the sign. And if t is too big, it would make the network more complex, and introduce more redundant noisy high-frequency eigenvectors. Here we vary t from 2 to 4. The table shows that $t = 3$ (Ours) achieves the best performance. Our input is defined on vertices. Although existing point-based deep learning methods such as PointNet [40] take extrinsic point coordinates as input, it is possible to feed the same intrinsic input to such architectures for prediction. So we also test this by feeding our input directly to PointNet [40] and report the accuracy of sign prediction. The performance is also lower than that of our method. This is probably due to our compact network design that generalizes well to new data.

TABLE 4
Evaluation of our design choices. We compare the accuracy (=number of correctly predicted signs/total number of eigenfunctions) of different design choices.

	Ours	Pos.	Diag.	2 Eig.	4 Eig.	PointNet
Acc.(%)	98.4	60.6	66.1	96.3	94.1	96.2

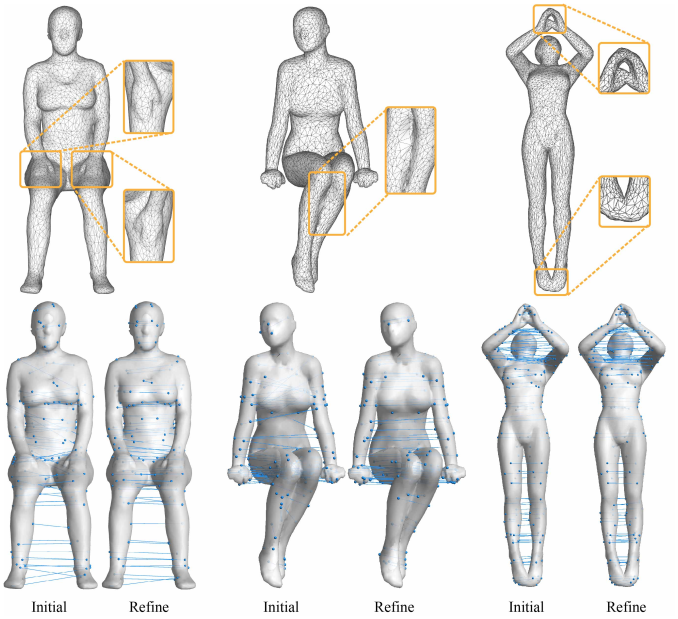


Fig. 7. Test with different topology. In this figure, we change the topology of models from SHREC and TOSCA to test the robustness of our method in difficult topology. We reconstruct the meshes by sticking spatially adjacent faces on the meshes together, so the shapes are no longer genus-zero. Meanwhile, the intrinsic symmetry correspondences obtained by our method are still reasonable. We also show both the initial symmetry and the refined symmetry in the second row. Since the reconstructed meshes are changed and are no longer perfectly intrinsically symmetric, the refinement step is important to polish the correspondences.

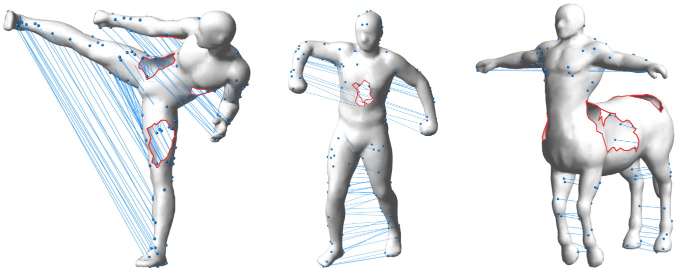


Fig. 8. Symmetry detection on incomplete surfaces.

5.4 Robustness

Different topology. Since the geodesic distance and the eigenfunctions are defined on the manifold $\mathcal{M}$, the topology of $\mathcal{M}$ would contribute significantly to the computation of intrinsic symmetry. For example, MT [7] requires the topology to be genus-zero. In our method, since the eigenfunctions can work consistently under different topology, the network can stay robust with topological changes. As shown in Figure 7, we reconstruct those meshes with self-intersection in space and the produced meshes are high-genus. The first row shows the original shapes with problematic regions highlighted. The second row shows the initial correspondences of intrinsic symmetry mapping and the correspondences after refinement. For those challenging cases, intrinsic symmetry is no longer precisely satisfied, and the refinement is effective in improving detected symmetry.

Incomplete shapes. Sometimes there could be missing data on shapes due to imperfect scanning or mesh modeling. We expect an intrinsic symmetry detection algorithm to work on such incomplete shapes. We perform a test by making some holes on the surface of the models. Figure 8 shows the results of our method. It can be seen that the symmetry pairs on the shapes are still reasonable.

Simple shape. We try to run intrinsic symmetry detection on a simple and regular shape as shown in Fig. 10, which is far from the training set. The code of [9], however, crashes because they rely on landmark pairs while there are more than two vertices with the same signature. In fact, [9] would crash on all such "regular" meshes with more than one symmetry. Our method

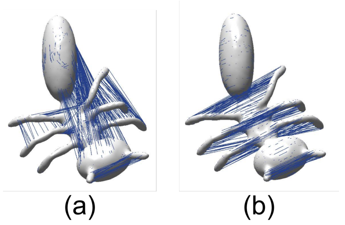
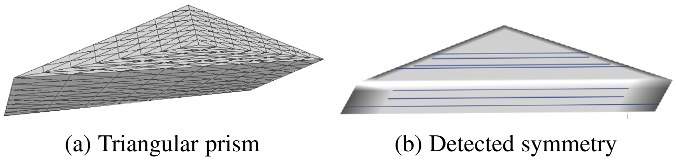


Fig. 9. An example with multiple symmetries. Our method (b) outperforms [9] (a) in the ant case where more than one symmetry exists. The up-to-down symmetry is not obvious but will confuse the landmark points finding algorithm in [9]. Our method can predict the symmetry correctly because it is based on the global properties of the eigenfunctions.



(a) Triangular prism

(b) Detected symmetry

Fig. 10. Symmetry detection on a simple triangular prism. This model is far from the training data, but our method can find the correct symmetry in (b). [9] crashes on this shape.

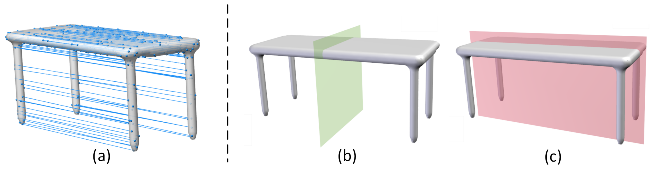


Fig. 11. A failure case of our method. (a) is the symmetry detection result of the 151st model in SHREC 2007 [35] computed by our method, (b) is the symmetry plane corresponding to our result, and (c) is another possible symmetry plane.

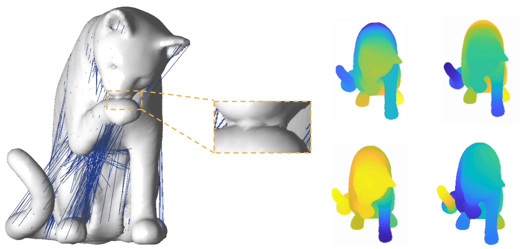


Fig. 12. A failure case when we glue the jaw and claw of a cat. The blue lines are the estimated symmetric point pairs. The eigenfunctions on the right are neither symmetric nor antisymmetric. The changed shape might not be considered as intrinsically symmetric by definition, and our method fails to find the intrinsic symmetry in this case.

works on the global property of eigenfunctions and thus is more robust. “Axiomatic” methods more or less have corner cases and require clean meshes, so robustness is another reason besides higher efficiency that we apply deep learning in intrinsic symmetry detection.

Multiple symmetries. Some shapes, including many creatures, actually have more than one intrinsic symmetry. The ant in Fig. 9 seemly has two symmetries, up-to-down and left-to-right. The up-to-down symmetry is not obvious but will confuse the landmark points finding algorithm in traditional intrinsic symmetry algorithms. As a comparison, [9] fails on this case, while our method can still predict the signs of eigenfunctions correctly, thus output a correct symmetry result, because it is based on the global properties of the eigenfunctions.

5.5 Failure Case

As shown by the statistics, our method works well in most cases. However, due to the deterministic network structure, our method can only predict one symmetry result for a certain object, even if it has multiple intrinsic symmetries. In Figure 11, the table has more than one reflectional symmetry plane, while our method cannot predict all of them. It would be our future work to extend our method to predict the entire symmetry group end-to-end.

Our method can also fail when the topology of an intrinsically symmetric shape changes too much. An example is shown in Figure 12 where the cat’s jaw and claw are glued together. The left side shows the shape and estimated correspondence. Our method fails for this case. The right side of the figure presents the 3rd-6th eigenfunctions on the shape. They are all neither symmetric nor antisymmetric, making our functional map based method fail. Actually, according to the definition, this new shape should no longer be considered as intrinsically symmetric.

6 Conclusions and Future Work

In this paper, we have presented a novel learning-based approach to intrinsic reflectional symmetry detection. Our method is based on functional maps and further develops a neural network architecture that predicts the sign of a Laplacian eigenfunction at a time. We design the network to take the first few Laplacian eigenfunctions, in addition to the eigenfunction to be predicted. Extensive experiments show higher efficiency and superior accuracy compared with state-of-the-art methods. We also performed experiments to validate design choices and robustness of our method in challenging cases.

This work addresses global intrinsic reflectional symmetry, which is most common in practice. Note that it can also detect some kind of rotational symmetries where there are only two elements in the rotation group. As future work, it would be interesting to also include rotational symmetry detection, although the property of the rotational symmetry functional map matrix is more complicated. Another possible direction is to extend this learning-based algorithm to partial symmetry detection. We would also provide the source code of this research work with Jittor implementation in the future, the Jittor is a fully just-in-time (JIT) compiled deep learning framework.

Acknowledgment

This work was supported by the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), the National Natural Science Foundation of China (No. 62061136007 and No. 61872440), Royal Society Newton Advanced Fellowship (No. NAF\R2\192151) and the Youth Innovation Promotion Association CAS.

References

[1] A. Tevs, Q. Huang, M. Wand, H.-P. Seidel, and L. Guibas, “Relating shapes via geometric symmetries and regularities,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 119, 2014.

[2] A. Dessein, W. A. Smith, R. C. Wilson, and E. R. Hancock, "Symmetry-aware mesh segmentation into uniform overlapping patches," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 95–107.

[3] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang, "Structure-aware shape processing," in *ACM SIGGRAPH 2014 Courses*. ACM, 2014, p. 13.

[4] P. Speciale, M. R. Oswald, A. Cohen, and M. Pollefeys, "A symmetry prior for convex variational 3d reconstruction," in *European Conference on Computer Vision*. Springer, 2016, pp. 313–328.

[5] N. J. Mitra, Y.-L. Yang, D.-M. Yan, W. Li, M. Agrawala *et al.*, "Illustrating how mechanical assemblies work," 2010.

[6] H. Wang and H. Huang, "Group representation of global intrinsic symmetries," in *Computer Graphics Forum*, vol. 36, no. 7. Wiley Online Library, 2017, pp. 51–61.

[7] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser, "Möbius transformations for global intrinsic symmetry analysis," in *Computer Graphics Forum*, vol. 29, no. 5. Wiley Online Library, 2010, pp. 1689–1700.

[8] M. Ovsjanikov, J. Sun, and L. Guibas, "Global intrinsic symmetries of shapes," in *Computer graphics forum*, vol. 27, no. 5. Wiley Online Library, 2008, pp. 1341–1348.

[9] R. Nagar and S. Raman, "Fast and accurate intrinsic symmetry detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 417–434.

[10] R. M. Rustamov, "Laplace-beltrami eigenfunctions for deformation invariant shape representation," in *Proceedings of the fifth Eurographics symposium on Geometry processing*. Eurographics Association, 2007, pp. 225–233.

[11] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, and Y. Xiong, "Partial intrinsic reflectional symmetry of 3d shapes," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, p. 138, 2009.

[12] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu, and B. Chen, "Multi-scale partial intrinsic symmetry detection," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 181, 2012.

[13] N. J. Mitra, L. J. Guibas, and M. Pauly, "Symmetrization," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3. ACM, 2007, p. 63.

[14] Y. Lipman, X. Chen, I. Daubechies, and T. Funkhouser, "Symmetry factored embedding and distance," in *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 103.

[15] D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Full and partial symmetries of non-rigid shapes," *International journal of computer vision*, vol. 89, no. 1, pp. 18–39, 2010.

[16] A. Dubrovina and R. Kimmel, "Matching shapes by eigendecomposition of the laplace-beltrami operator," in *3DPVT*, 2010.

[17] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov, "Zoomout: spectral upsampling for efficient shape correspondence," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 155:1–155:14, 2019. [Online]. Available: https://doi.org/10.1145/3355089.3356524

[18] T. Shao, Y. Yang, Y. Weng, Q. Hou, and K. Zhou, "H-CNN: spatial hashing based CNN for 3d shape analysis," *CoRR*, vol. abs/1803.11385, 2018. [Online]. Available: http://arxiv.org/abs/1803.11385

[19] A. Poulenard, M. Rakotosaona, Y. Ponty, and M. Ovsjanikov, "Effective rotation-invariant point CNN with spherical harmonics kernels," in *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*. IEEE, 2019, pp. 47–56. [Online]. Available: https://doi.org/10.1109/3DV.2019.00015

[20] R. Huang, M. Rakotosaona, P. Achlioptas, L. J. Guibas, and M. Ovsjanikov, "Operatornet: Recovering 3d shapes from difference operators," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 8587–8596. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00868

[21] R. Luo, T. Shao, H. Wang, W. Xu, X. Chen, K. Zhou, and Y. Yang, "Nnwarp: Neural network-based nonlinear deformation," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 4, pp. 1745–1759, 2020. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2881451

[22] X. Chen, Y. Li, X. Luo, T. Shao, J. Yu, K. Zhou, and Y. Zheng, "Autosweep: Recovering 3d editable objects from a single photograph," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 3, pp. 1466–1475, 2020. [Online]. Available: https://doi.org/10.1109/TVCG.2018.2871190

[23] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 3189–3197.

[24] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proceedings of the IEEE international conference on computer vision workshops*, 2015, pp. 37–45.

[25] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[26] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[27] L. Yi, H. Su, X. Guo, and L. J. Guibas, "SyncSpecCNN: Synchronized spectral CNN for 3d shape segmentation." in *CVPR*, 2017, pp. 6584–6592.

[28] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5659–5667.

[29] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov, "Unsupervised deep learning for structured shape matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1617–1627.

[30] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 8589–8598. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.00862

[31] M. Li, D. M. Kaufman, V. G. Kim, J. Solomon, and A. Sheffer, "Optcuts: joint optimization of surface cuts and parameterization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–13, 2018.

[32] O. Halimi, O. Litany, E. Rodola, A. M. Bronstein, and R. Kimmel, "Unsupervised learning of dense shape correspondence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4370–4379.

[33] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, "Functional maps: a flexible representation of maps between shapes," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 30, 2012.

[34] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," pp. 35–57, 2003.

[35] D. Giorgi, S. Biasotti, and L. Paraboschi, "Shrec: shape retrieval contest: Watertight models track," *Online]: http://watertight. ge. imati. cnr. it*, vol. 7, 2007.

[36] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004.

[37] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: shape completion and animation of people," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 408–416.

[38] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[39] F. Bogo, J. Romero, M. Loper, and M. J. Black, "Faust: Dataset and evaluation for 3d mesh registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3794–3801.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] V. G. Kim, Y. Lipman, and T. Funkhouser, "Blended intrinsic maps," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 79.

[43] X. Liu, S. Li, R. Liu, J. Wang, H. Wang, and J. Cao, "Properly constrained orthonormal functional maps for intrinsic symmetries," *Computers & Graphics*, vol. 46, pp. 198–208, 2015.

[44] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis, "The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces," in *Advances in neural information processing systems*, 2005, pp. 33–40.

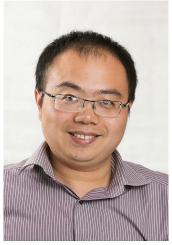
[45] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," vol. 27, no. 3, p. 97, 2008.

Yi-Ling Qiao received a bachelor's degree in computer science and technology from the University of Chinese Academy of Sciences in 2019. He is currently a PhD student in computer science at University of Maryland, College Park. His research interests include computer graphics and geometric processing.

Lin Gao received his PhD degree in computer science from Tsinghua University. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young researcher award. His research interests include computer graphics and geometric processing.

Shu-Zhi Liu is currently an undergraduate student in computer science and technology at the University of Chinese Academy of Science. His research interests includes computer graphics.

Ligang Liu received the BSc and PhD degrees from Zhejiang University, China, in 1996 and 2001, respectively. He is a professor with the University of Science and Technology, China. Between 2001 and 2004, he was at Microsoft Research Asia. Then he was at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. His research interests include geometric processing and image processing. His research works could be found at his research website: http:// staff.ustc.edu.cn/ lgliu. He is a member of the IEEE.



Yu-Kun Lai received his bachelor's degree and PhD degree in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Professor in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial boards of *Computer Graphics Forum* and *The Visual Computer*.

Xilin Chen received the BS, MS, and PhD degrees in Computer Science from Harbin Institute of Technology, China, in 1988, 1991, and 1994 respectively. Then he joined the Department of Computer Science and Engineering, Harbin Institute of Technology, where he was a lecturer (1994), associate professor (1996), and professor (1999-2005). He was a visiting scholar with Carnegie Mellon University from 2001 to 2004. He was elected into the 100 Talents Program of Chinese Academy of Sciences (CAS) and joined the Institute of Computing Technology, CAS in August, 2004. His research interests are Image Understanding, Computer Vision, Pattern Recognition, Image Processing, Multimodal Interface, and Digital Video Broadcasting.